# Immers

## **微信登录/绑定**

1、

**需要获得所开发的项目在微信申请的：*AppID、AppSecret***

2、

*//导入微信登录所需的依赖包*implementation **'com.tencent.mm.opensdk:wechat-sdk-android:+'**

**3、**

创建包wxapi，在包内添加WXAuthorization类和微信登录入口页面WXEntryActivity类

WXAuthorization.class：

在WXAuthorization(Activity）方法中进行**创建WXAPI传入本项目的AppID，注册APP和广播接收。**

在此操作之后需检测用户是否安装微信

如果已安装，则将SendAuth.Req req数据传至WXEntryActivity。

WXEntryActivity.class：

**创建WXAPI传入本项目的AppID，接收api传过来的数据**

onResp()方法用于接收授权通知，在用户同意授权后，通过AppID和AppSecret以及授权的authResp.code，Get请求获取AccessToken和OpenId，获取到的微信用户信息可通过事件通知发出去，在对应的页面进行接收。

4、

**在需要进行微信登录或绑定的页面调用**WXAuthorization(this);

注册事件通知，接收微信用户信息。

## **通过阿里云存储OSS上传用户头像**

**需要获得所开发的项目在阿里云申请的：*ALIYUN\_OSS\_ENDPOINT、BUCKET\_NAME、releaseObjectKey***以及***headerImage***

1、

通过后台接口获取accessKeyId、accessKeySecret以及securityToken

2、创建credentialProvider

OSSCredentialProvider credentialProvider = new OSSStsTokenCredentialProvider(accessKeyId, accessKeySecret, securityToken);

3、

进行超时设置ClientConfiguration conf;

4、

将endpoint和超时设置传入OSS

OSS oss = **new** OSSClient(getApplicationContext(),***ALIYUN\_OSS\_ENDPOINT***, credentialProvider, conf);

5、

将***BUCKET\_NAME***、***releaseObjectKey***以及要上传的图片路径**path**传入PutObjectRequest put

6、

***headerImage***+put.getObjectKey();就是上传到阿里云的头像的Uri

## MQTT

在Application中先初始化

MqttManager.getInstance().setContext(this);

在Main的oncreate中调用MqttService的connectHandler()进行Mqtt的连接。

MqttService:

MqttManager.*getInstance*().connect(**new** ConnectCommand()，**new** IMqttActionListener());

**new** ConnectCommand()设置IP、端口、超时以及用户名称密码等

**new** IMqttActionListener()监听Mqtt的连接状态

3、

当Mqtt连接成功之后开始订阅主题（Qos—质量，topic—主题）

MqttManager.*getInstance*().sub(**new** SubCommand().setQos(0).setTopic(topic), **new** IMqttActionListener())

4、

在需要接收数据的页面调用MqttService的receiveData()

进行监听主题。

MqttService：

MqttManager.*getInstance*()  
 .registerMessageListener(**new** MqttCallback());

## EventBus事件

1、

//添加依赖

*//用来传递消息*implementation **'org.greenrobot:eventbus:3.0.0'**

2、

*//在需要订阅时间的地方注册事件(一般在页面的oncreate就进行注册)*EventBus.*getDefault*().register(**this**);

*//取消事件订阅（注册完一定要记得在onDestory（）中取消订阅）*

EventBus.getDefault().unregister(this);

3、

发送消息

EventBus.*getDefault*().post(Object event);

4、

接收消息

@Subscribe(threadMode = ThreadMode.***MAIN***)  
**public void** onMessageEvent(Object event){  
   
}